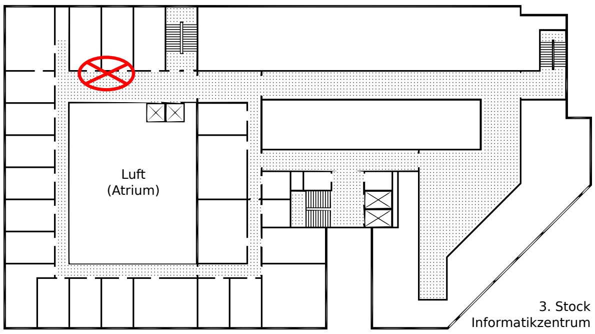
Abteilung Algorithmik **Winter 2014/15**
Institut für Betriebssysteme und Rechnerverbund
TU Braunschweig

Prof. Dr. Sándor P. Fekete
Dr. Christian Scheffer

Algorithmen und Datenstrukturen
Übung 3 vom 12. 12. 2014

Abgabe der Lösungen bis zum Donnerstag, den 8. 1. 2015 um 14:30 im Hausaufgabenrückgabeschrank.

Bitte die Blätter zusammenheften und vorne deutlich mit eigenem Namen, Matrikel- und Gruppennummer, sowie Studiengang versehen!



Aufgabe 1 ($\mathcal{O}$-Notation): In der Vorlesung wurde die $\mathcal{O}$-Notation zur Abschätzung der Laufzeit eingeführt.

a) Suche für die folgenden Funktionen jeweils geeignete Konstanten c sowie n_0 und zeige mit Hilfe dieser Konstanten, dass die jeweilige Funktion in der angegebenen Klasse liegt.

$$\begin{array}{rcrcl} f_1(n) & = & \frac{1}{n^2} & \in & \mathcal{O}(1) \\ f_2(n) & = & 10n^4 + 5n^2 & \in & \mathcal{O}(n^4) \\ f_3(n) & = & 111n^2 & \in & \mathcal{O}(n^3) \\ f_4(n) & = & \log_2 n & \in & \mathcal{O}(\log_{10} n) \\ f_5(n) & = & n^2 - \log(n) \cdot n & \in & \mathcal{O}(n^2) \end{array}$$

b) Kreuze an, in welchen Klassen die jeweilige Funktion liegt (ohne Begründung).

$f(n)$	$\Omega(1)$	$\Theta(1)$	$\mathcal{O}(1)$	$\Omega(n)$	$\Theta(n)$	$\mathcal{O}(n)$	$\Omega(n^2)$	$\Theta(n^2)$	$\mathcal{O}(n^2)$
10									
$2n^2 + \log n$									
1.01^{1000}									
$2\sqrt{n}\log n$									
$3.14n$									
$3\log(n^2) + 1$									

(10+6 Punkte)

Aufgabe 2 ($\mathcal{O}$-Notation): Im Folgenden sollen die Definitionen der $\mathcal{O}$-Notationen aus der Vorlesung verwendet werden. Beweise Satz 3.12 aus der Vorlesung: Seien $f, g : \mathbb{N} \to \mathbb{R}$ zwei Funktionen. Dann gilt:

a) $f \in \Theta(g) \Leftrightarrow g \in \Theta(f)$

b) $f \in \Theta(g) \Leftrightarrow f \in \mathcal{O}(g)$ und $f \in \Omega(g)$

c) $f \in \mathcal{O}(g) \Leftrightarrow g \in \Omega(f)$

(5+5+5 Punkte)

Aufgabe 3 (Laufzeitkomplexität von Algorithmen): Du hast einen Input von n Elementen und einen Algorithmus, der einzelne Elemente bearbeitet. Das kostet $n^3 + 11 \log n$ Operationen pro Element. Alle Elemente müssen bearbeitet werden.

a) Wie lange dauert das (mit und ohne $\mathcal{O}$-Notation)?

b) Die Bearbeitung zweier Elemente ist unabhängig voneinander, man kann sie also parallelisieren. Dazu stehen 5 Kerne zur Verfügung. Wie wirkt sich dies auf die Gesamtlaufzeit aus (mit und ohne $\mathcal{O}$-Notation)?

c) Alternativ kannst du ein Preprocessing verwenden, welches in $2n^{1.5} - 3n + 71$ läuft und danach die Bearbeitungszeit pro Element auf $n \log n$ reduziert, allerdings ist die Bearbeitung der Elemente dann nicht mehr parallelisierbar. Wie wirkt sich das auf die Gesamtlaufzeit aus (mit und ohne $\mathcal{O}$-Notation)?

d) Für welche Möglichkeit (ursprünglicher Algorithmus, b), oder c)) entscheidest du dich, um die Leistungsfähigkeit des Programms für große Inputmengen zu maximieren? Begründe deine Wahl.

(4+4+5+4 Punkte)

Aufgabe 4 (Komplexitätsklassen): Sortiere die folgenden Funktionsklassen nach Inklusion (ohne Begründung). Kennzeichne identische Klassen.

$$\mathcal{O}(n^{10} - n),\ \mathcal{O}(\log_2(n^5)),\ \mathcal{O}(3^n),\ \mathcal{O}(7\sqrt{n}),\ \mathcal{O}(n \log_2 15),\ \mathcal{O}(19),$$

$$\mathcal{O}(3n^2 + 7),\ \mathcal{O}(2^n),\ \mathcal{O}(1),\ \mathcal{O}(n^5),\ \mathcal{O}(n),\ \mathcal{O}(\sqrt{n})$$

(Hinweis: Die Mengen $A = \{1,2,3\}$, $B = \{1,2\}$, $C = \{1,2,3,4\}$ und $D = \{1,2,3\}$ nach Inklusion sortiert sind: $B \subseteq A = D \subseteq C$.) **(12 Punkte)**